第8章　SABRモデルと最適化法

この章ではボラティリティのスマイル/スキュー (以下スキューは省略) を表現するモデルについて説明する。まず CEV モデルのスマイルの表現法を確認後、パラメータをスワップションスマイルにキャリブレート (calibrate) する。次に CEV モデルのボラティリティを確率的に変動させる SABR モデルを説明し、各パラメータの特徴を確認する。最後にマイナス金利に対応させる為、ノーマルボラティリティと Shifted SABR モデルでの数値例を紹介する。
尚、随時キャリブレートに必要な最適化法を例示する。

8.1　CEV モデル

初めにこれまで説明したモデルを復習しよう[*1]。t 時点の無配当株式の価格を S_t で表すと、1973 年のブラックショールズモデルでは次式で価格の変動を表した。

$$\begin{aligned} dS_t &= rS_t\ dt + \sigma_B S_t\ dW_t \\ & \quad r: \text{リスクフリーレート}\ , \quad \sigma_B: \text{ブラックボラティリティ} \\ & \quad W: \text{標準ブラウン運動} \end{aligned}$$

1976 年のブラックモデルでは S_t のフォワード価格を F_t とし、次式のようにドリフト項が無い式となった。

$$dF_t \;=\; \sigma_B F_t\ dW_t \quad \left(=\ \sigma_B F_t^1\ dW_t\right) \tag{8.1}$$

F_t には債券先物価格やフォワードスワップ金利等も含まれ、以下では単に**フォワード $\boldsymbol{F_t}$** として参照する。

一方ノーマルモデルでは σ_N をノーマルボラティリティとして、

$$dF_t \;=\; \sigma_N\ dW_t \quad \left(=\ \sigma_N F_t^0\ dW_t\right) \tag{8.2}$$

と表される。これらの式のカッコ内に記した F_t の肩にある 1 と 0 は“1 乗”, “0 乗”の**指数** (exponent) を意味している。

[*1] ラフな説明の為、ニューメレールは無視している。

8.1.1　CEV モデルのイメージ

この指数部分を変数 β で置き換えたモデルが先物やフォワード契約用の **CEV モデル** (**CEV model**: Constant Elasticity of Variance, または "セブ") である。

$$dF_t \;=\; \alpha F_t^{\beta}\, dW_t \qquad (0 < \alpha,\ 通常\ 0 \le \beta \le 1) \tag{8.3}$$

α は CEV モデルのボラティリティを意味し、正の定数である。

- $\beta = 1$ の場合 CEV モデルはブラックモデルとなるので、$\alpha = \sigma_B$ であり、F_t は対数正規分布を仮定。
- $\beta = 0$ の場合 CEV モデルはノーマルモデルとなり $\alpha = \sigma_N$ で、F_t は正規分布を仮定。

つまり CEV モデルは β の値の応じ、対数正規分布や正規分布、その中間の分布を仮定することが出来る。[*2]

8.2 節で説明する SABR モデル (Stochastic α β ρ) は名前が示すように式 (8.3) のボラティリティー α を確率変動させるモデルであり、始めに α が定数である CEV モデルの説明をしよう。

表 8.1: 2Y×5Y JPY スワップション スマイル　(データ出所:Bloomberg)

	-50bp	-25bp	ATM	25bp	50bp	100bp	200bp
Strike(%)	0.06	0.31	0.56	0.81	1.06	1.56	2.56
Black Vol(%)	90.78	46.09	45.3	50.17	53.85	58.42	62.72

(200bp 列 Strike 計算例: 200bp+ATM0.56%=2.56)

表 8.1 は 2014 年 9 月 15 日の 2 年満期でテナー 5 年の円スワップションのスマイルである。ATM のスワップレートは 0.56%、ブラックボラティリティが 45.30% [*3] であり、$-50bp$ から $+200bp$ まで 7 つのボラティリティの建値となっている。

図 8.1 は $-50bp$ 列のストライク 0.06%, ボラティリティ 90.78%の建値を除外して、CEV モデルのパラメーター α 及び β に対してキャリブレートを行った曲線である。

キャリブレートされた 2 つのパラメータの値は図の右上に表示された $\alpha = 2.67$、$\beta = 1.33$ で、式 (8.3) に当てはめると、

$$dF_t \;=\; 2.67 \times F_t^{1.33}\, dW_t$$

[*2] 特に $\beta = 0.5$ とし、ドリフト項に平均回帰性を持たせたショートレートモデルは **CIR モデル** (Cox Ingersoll Ross) と呼ばれ、$\beta = 0.5$ (0.5 乗) から Square-root process として参照される。CIR モデルの数値例は Clewlow *et al*. [19] を参照。

[*3] 当時はノーマルボラティリティへの移行期であり、ブラックボラティリティがスワップションで建値されていた。

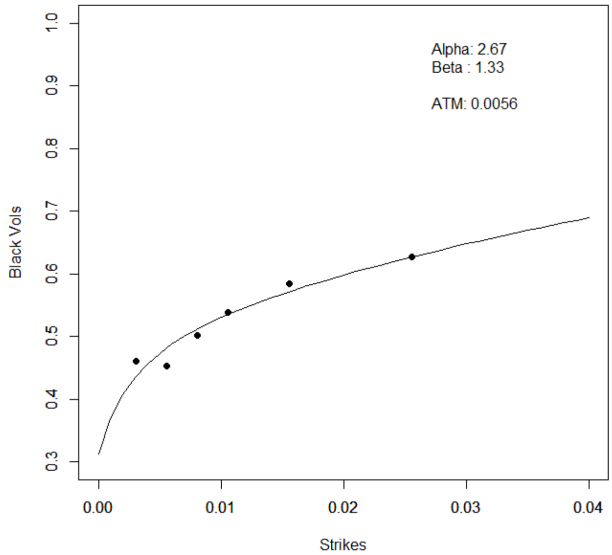


図 8.1: $-50bp$ 列を除く 6 個のデータでのキャリブレート

のモデルとなっている。この図から CEV モデルでは、例えば 4%行使価格のスワップションのブラックボラティリティは 70%程度となることを示している。

このパラメータの当てはめから理解すべきことは、CEV モデルでは 2 つのパラメータがあれば、スマイルを表現出来るモデルとなっている点である。CEV のボラティリティ($\alpha = 2.67$) は定数であるにも係らず、行使価格に応じてブラックのボラティリティ スマイルに近い値が計算されている。

従って、CEV モデルではブラックモデルのように行使価格毎にボラティリティを変更する必要が無い。

8.1.2 ブラックボラティリティ近似式

ではこのパラメータを使用し、行使価格 4%のブラックのボラティリティを計算しよう。それには 1999 年 Hagan and Woodward [28] の“Equivalent Black volatilities”で発表した以下のブラックと CEV モデルのボラティリ

ティーの関係式を使用する。(この章では同 [28] 及び [29] に倣い、行使価格は K で表す)

$$\overbrace{\sigma_B}^{\text{ブラック ボラティリティ}} = \underbrace{\frac{\alpha}{F_k^{1-\beta}}}_{1\text{ 項目}}\left\{1+\underbrace{\frac{(1-\beta)(2+\beta)}{24}\left(\frac{F-K}{F_k}\right)^2}_{2\text{ 項目}}+\underbrace{\frac{(1-\beta)^2}{24}\frac{\alpha^2 T}{F_k^{2-2\beta}}}_{3\text{ 項目}}+\cdots\right\} \tag{8.4}$$

$$F：\text{現在のフォワード},\quad K：\text{行使価格},\quad T：\text{満期年}$$

$$F_k \;:=\; \frac{1}{2}(F+K),\qquad \alpha\,,\;\beta\;:\;CEV\ \text{モデルのパラメータ}$$

この式を使用し、図 8.2 では 3 行目の行使価格に対して、12 行目でブラックのボラティリティを算出した。例えば C3 セルで行使価格 4%の場合、C12 セルではブラックボラティリティ 0.69088(69.088%) が計算される。簡単にシートを説明しよう。

	A	B	C	D
1	CEVモデルのBlackボラティリティ			
2				
3	行使価格:K	0.01	0.04	C列計算式
4	ATM: Fo	0.0056	0.0056	
5	満期: T	2	2	
6	CEV Vol: α	2.67	2.67	キャリブレーションで求める
7	β	1.33	1.33	キャリブレーションで求める
8	Fk	0.0078	0.0228	=(C4+C3)/2
9	1項目	0.53815	0.76671	=C6/C8^(1-C7)
10	2項目	-0.01457	-0.10423	=(1-C7)*(2+C7)/24*((C4-C3)/C8)^2
11	3項目	0.00263	0.00533	=(1-C7)^2/24*(C6^2*C5)/C8^(2-2*C7)
12	Black Vol	0.53172	0.69088	=C9*(1+C10+C11)

図 8.2: 式 (8.4) でのブラックボラティリティ

- 6, 7 行目で $\alpha = 2.67$、$\beta = 1.33$ とし 式 (8.4) を計算。
- 式中の“1 項目”、“2 項目”、“3 項目”が図 8.2 の 9 行目から 11 行目に対応。
- このエクセルの行使価格を 0%から 4%まで 0.01%刻みでブラックボラティリティを算出し、図 8.1 を描写。

図 8.2 から解かるように 式 (8.4) の計算は見慣れない係数が多い割りに単純である。ただ、この式の導出は漸近展開 (Asymptotic expansion) や特異摂動法 (Singular perturbation theory) 等、かなり高度な知識が要求される為、本書では利用法の記述のみとする。

尚、式 (8.4) 右辺最後の“$\cdots$”部分は漸近展開した際の小さな項目がまだ続くことを意味しているが、通常この部分は計算の対象外とされる。

8.1.3　モデル キャリブレート

8.1.2 節で使用した $\alpha = 2.67$、$\beta = 1.33$ の求め方、つまり表 8.1 のスワップション スマイルから CEV モデルの**キャリブレーション** (パラメータの推定) を説明しよう。

その方法は市場で建値されたスワップションのボラティリティ(**マーケット ボラティリティ**: $\boldsymbol{\sigma^{mkt}}$) と式 (8.4) で計算されたブラックのボラティリティ(**モデル ボラティリティ**: $\boldsymbol{\sigma^{model}}$) との差を最小にするようなパラメータ α、β を見つけることである。この計算を**最適化**と言う。

“最小”を計算する式を**目的関数**と呼ぶが、**平均 2 乗誤差** (**MSE**:Mean Squared Error) や **2 乗平均平方根誤差** (**RMSE**:Root Mean Squared Error) 等が目的関数として使われる。

平均 2 乗誤差 ($\boldsymbol{MSE}$)：

$$\frac{1}{N}\sum_{i=1}^{N}\Big(\underset{\text{マーケット}}{\sigma_i^{mkt}} - \underset{\text{モデル}}{\sigma_i^{model}}\Big)^2$$

2 乗平均平方根誤差 ($\boldsymbol{RMSE}$)：

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}\Big(\underset{\text{マーケット}}{\sigma_i^{mkt}} - \underset{\text{モデル}}{\sigma_i^{model}}\Big)^2} \tag{8.5}$$

図 8.3 で上下に 2 つエクセルを配置したが、上側がキャリブレート前で α と β (C2 セルと E2 セル) はともにゼロの状態であり、下側がキャリブレート後で $\alpha = 2.6711$ と $\beta = 1.3306$ を算出。

G16 セルで目的関数 (RMSE) の計算が行われているが、キャリブレート前の値が 0.53135 であり、$\alpha = 2.6711$、$\beta = 1.3306$ で最小値 0.01690 となっている。

この値の算出は **Excel ソルバー**機能[*4] を使用したが、図 8.3 の中央のスクリーンショットはソルバーに与えたセル情報や最適化で使用したアルゴリズム等である。では図 8.3 の数値を見て行く。

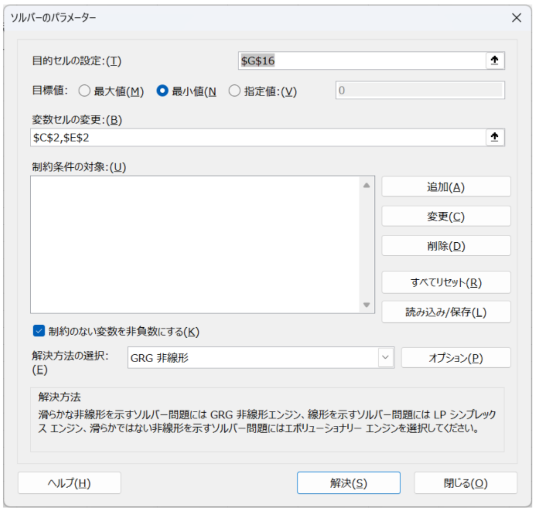
3 行: 行使価格： B3〜G3 セルの行使価格は表 8.1Strike(%) 行の値を実数で入力。($-50bps$ 列の“0.06”は除外したので、$-25bps$ から $200bps$ までの 6 つの値)

6, 7 行 $\boldsymbol{\alpha}$、$\boldsymbol{\beta}$： 6, 7 行目の α、β は C2 セルと E2 セルを参照させ、共に初期値はゼロ。

12 行: Black Vol (**モデル ボラティリティ $\boldsymbol{\sigma^{model}}$**)**：** 式 (8.4) を使用し、与えられた行使価格毎にブラックボラティリティを計算。(計算は図 8.2 と同じ)

[*4] Excel ソルバー機能はアドインとしてインストールする必要がある。インストール後は“データ”タブに表示される。

	A	B	C	D	E	F	G	H
1	CEVのα βのキャリブレーション							
2		α＝	0.0000	β＝	0.0000			(前図からの変更点)
3	行使価格:K	0.0031	0.0056	0.0081	0.0106	0.0156	0.0256	G列計算式
4	ATM: Fo	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056	
5	満期: T	2	2	2	2	2	2	
6	CEV Vol:α	0	0	0	0	0	0	=$C$2
7	β	0	0	0	0	0	0	=$E$2
8	Fk	0.00435	0.0056	0.00685	0.0081	0.0106	0.0156	
9	1項目	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	
10	2項目	0.02752	0.00000	0.01110	0.03175	0.07417	0.13697	
11	3項目	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	
12	Black Vol	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	
13	market Vol	0.46090	0.45300	0.50170	0.53850	0.58420	0.62720	
14	差の２乗	0.21243	0.20521	0.25170	0.28998	0.34129	0.39338	=(G13-G12)^2
15						平均2乗誤差(MSE)	0.28233	=SUM(B14:G14)/6
16						2乗平均平方根誤差(RMSE)	0.53135	=G15^0.5



	A	B	C	D	E	F	G	H
1	CEVのα βのキャリブレーション							
2		α＝	2.6715	β＝	1.3306			(前図からの変更点)
3	行使価格:K	0.0031	0.0056	0.0081	0.0106	0.0156	0.0256	G列計算式
4	ATM: Fo	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056	
5	満期: T	2	2	2	2	2	2	
6	CEV Vol:α	2.67153	2.67153	2.67153	2.67153	2.67153	2.67153	=$C$2
7	β	1.33063	1.33063	1.33063	1.33063	1.33063	1.33063	=$E$2
8	Fk	0.00435	0.00560	0.00685	0.00810	0.01060	0.01560	
9	1項目	0.44256	0.48110	0.51424	0.54355	0.59410	0.67507	
10	2項目	-0.01516	0.00000	-0.00611	-0.01748	-0.04084	-0.07542	
11	3項目	0.00178	0.00211	0.00241	0.00269	0.00322	0.00415	
12	Black Vol	0.43664	0.48212	0.51234	0.53551	0.57175	0.62696	
13	market Vol	0.46090	0.45300	0.50170	0.53850	0.58420	0.62720	
14	差の２乗	0.00059	0.00085	0.00011	0.00001	0.00015	0.00000	=(G13-G12)^2
15						平均2乗誤差(MSE)	0.00029	=SUM(B14:G14)/6
16						2乗平均平方根誤差(RMSE)	0.01690	=G15^0.5

図 8.3: Excel ソルバーによるキャリブレート例

(上: キャリブレート前、下: キャリブレート後)

- 算出されるボラティリティが式 (8.5) のモデル ボラティリティ σ_i^{model}。

13 行: market Vol (マーケット ボラティリティ σ^{mkt}) : 表 8.1 の Black Vol の値を実数で入力し、market Vol とする。

G16: RMSE の計算 : 14 行と G15 セルから式 (8.5) を算出。

14 行: 各行使価格別に $(\sigma_i^{mkt} - \sigma_i^{model})^2$ を計算。

G15: 14 行目の計算を基に平均 2 乗誤差 (MSE) を算出。

図 8.3 中央: ソルバー画面 :

最適化の実行として、G16 セルを最小にする α($C2$ セル)、β($E2$ セル) の値をソルバーで見つける。

- 次のセル情報をソルバーに設定し、" 解決 " をクリック。

目的セル	:	$G16$, 目標値 : 最小値
変数セル	:	$C2$ と $E2$, 制約のない変数は非負数
アルゴリズム	:	GRG 非線形

図 8.3 下側: C2:($\alpha = 2.6711$), E2:($\beta = 1.3306$) :

ソルバーによって最適化され、C2 セル ($\alpha = 2.6711$) と E2 セル ($\beta = 1.3306$) を算出。

以上が CEV モデルの α, β の算出方法となる。図 8.3 中央のソルバーで選択した **GRG** とは Generalized Reduced Gradient 一般化簡約勾配法を意味するが、Microsoft 社 (または Excel ソルバーのコードを提供したと思われる Frontline Systems 社) からの詳細な計算法の開示は無いようである。[*5]

ただし Excel ソルバーでは解が得られない場合があり、" 最適化 " に関しては Python 等でダブルチェック出来るようになるべきであろう。(8.2.3 節で Python の最適化コードを紹介)

尚 QuantLib-Python で CEV モデルを扱うクラスは無い為、Python のコード例は割愛する。(QuantLib の C++は `CEVCalculator` クラスを提供)

($-50bp$ 列を加えたキャリブレート)

キャリブレートの方法が判ったところで、図 8.4 を見てみよう。この図は当初除外した $-50bp$ 列 (行使価格 0.06% ボラティリティ 90.78%) のデータを加えて、

[*5] Frontline Systems 社の資料では"The GRG method can be viewed as a nonlinear extension of the Simplex method, which selects a basis, determines a search direction, and performs a line search on each major iteration, solving systems of nonlinear equations at each step to maintain feasibility."との記述がある。

	A	B	C	D	E	F	G	H	I
1	CEVのα βのキャリブレーション								
2		α＝	0.1594	β＝	0.7476				
3	行使価格:K	0.0006	0.0031	0.0056	0.0081	0.0106	0.0156	0.0256	H列計算式
4	ATM: Fo	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056	
5	T	2	2	2	2	2	2	2	
6	CEV Vol :	0.1594	0.1594	0.1594	0.1594	0.1594	0.1594	0.1594	=$C$2
7	β	0.7476	0.7476	0.7476	0.7476	0.7476	0.7476	0.7476	=$E$2
8	fk	0.0031	0.00435	0.0056	0.00685	0.0081	0.0106	0.0156	
9	1項目	0.68497	0.62883	0.58999	0.56074	0.53751	0.50223	0.45556	
10	2項目	0.07517	0.00954	0.00000	0.00385	0.01101	0.02572	0.04749	
11	3項目	0.00249	0.00210	0.00185	0.00167	0.00153	0.00134	0.00110	
12	Black76 Vo	0.73816	0.63615	0.59108	0.56383	0.54425	0.51582	0.47770	
13	market Vol	0.90780	0.46090	0.45300	0.50170	0.53850	0.58420	0.62720	
14	差の２乗	0.02878	0.03071	0.01907	0.00386	0.00003	0.00468	0.02235	=(H13-H12)^2
15							平均2乗誤差(MSE)	0.01564	=SUM(B14:H14)/7
16							2乗平均平方根誤差(RMSE)	0.12506	=H15^0.5

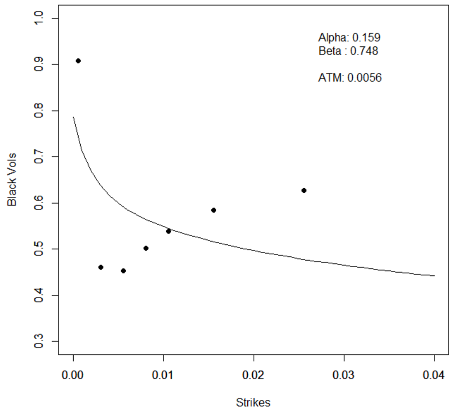


図 8.4: 行使価格 0.06% ボラティリティ 90.78%を追加した CEV モデル再度キャリブレートを行ったものである。[*6]

$\alpha = 0.159$ と $\beta = 0.748$ とそれらしいパラメータが算出出来ているが、ブラックのボラティリティを描いた実線は実用に耐えないものとなっている。

このようなスマイルにも対応できるモデルが次節の SABR モデルである。

8.2　SABR モデル

SABR モデル (Stochastic α β ρ, セイバー) とは名前が示すように CEV モデルのボラティリティー α を確率変動させるモデルである。

図 8.5 は図 8.4 のボラティリティデータに対し、SABR モデルを使用してキャリブレートし、ブラックのボラティリティを算出させたグラフである。

図 8.4 と比較すると SABR モデルが優れていることは一目瞭然であり、ボラティリティを補間/補外する際、標準的に使用されているモデルとなっている。

[*6] もし GRG 非線形で解が得られない場合、“エボリューショナリー”で試そう。

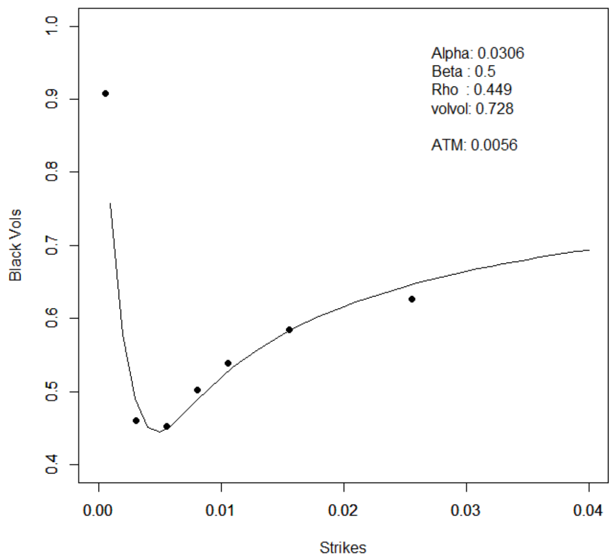


図 8.5: SABR のキャリブレート結果とブラックボラティリティ

図 8.5 の右上にはキャリブレートされた 4 つのパラメータが表示されているが、CEV モデルと比べ、Rho(ρ: ロー) と volvol(ボルボルまたは ν: ニュー) の 2 つが増えている。

$\boldsymbol{\rho}$(ロー) はフォワード F_t とボラティリティ α_t の相関係数であり、**volvol($\boldsymbol{\nu}$ ニュー)** はボラティリティ α_t のボラティリティなので $volvol$(ボルボル) と呼ばれる場合が多い。

では SABR モデルの次の 3 つの方程式を CEV モデルの式 (8.3) と比較しながら見ていく。

$$dF_t = \alpha_t F_t^{\beta}\, dW_t^1\,, \qquad (\text{通常 } 0 < \beta \leq 1) \tag{8.6}$$

$$d\alpha_t = \underbrace{\nu}_{\substack{\text{ボル}\\\text{ボル}}} \alpha_t\, dW_t^2\,, \qquad (0 < \nu\,,\ 0 < \text{初期値 } \alpha_0) \tag{8.7}$$

$$dW_t^1\, dW_t^2 = \underbrace{\rho}_{\substack{\text{相関}\\\text{ロー}}}\, dt\,, \qquad (-1 \leq \rho \leq 1) \tag{8.8}$$

1 つ目の式 (8.6) は CEV の式 (8.3) とほとんど同じである。異なる点はボラティリティ α が時間の関数として、α_t となっている部分であるが、この α_t の動き方を式 (8.7) と (8.8) が説明する。

式 (8.7) の α_t を F_t に置き換えれば、ブラックモデルの式 (8.1) と同じとなることから、SABR モデルはボラティリティ α_t が対数正規分布していることを仮定している。従って、α_t の対数をとった標準偏差 (年率化) がボルボル ν である。

3 番目の式 (8.8) は 2 つの確率変数、フォワード F_t とそのボラティリティ

α_t の相関を記述する一般的な方法で、特に深い意味は無く、ρ(ロー) というパラメータが相関係数であると理解すれば 十分である。尚 相関係数なので、$-1 \le \rho \le 1$ の範囲に限定。

8.2.1　ブラックボラティリティ近似式

次にこれら 4 つのパラメータのキャリブレート ($\beta = 0.5$ に設定するのが一般的で、パラメータは 3 つ) を CEV モデルで行った同様の分析手法で説明しよう。SABR モデルでのブラックボラティリティーの近似式は Hagan *et al*. [29] によって次のように発表[*7] されている。(各記号は式 (8.4) に同じ)

$$\underset{\sigma_B}{\overset{\text{ブラック ボラティリティ}}{}} = \frac{z}{x(z)} \times \frac{\alpha_0 \Big\{ 1 + \Big[\overbrace{\frac{(1-\beta)^2}{24} \frac{\alpha_0^2}{(FK)^{1-\beta}}}^{\text{分子 2 項}} + \overbrace{\frac{\alpha_0 \beta \rho \nu}{4(FK)^{(1-\beta)/2}}}^{\text{分子 3 項}} + \overbrace{\frac{2-3\rho^2}{24}\nu^2}^{\text{分子 4 項}} \Big] T + \cdots \Big\}}{(FK)^{(1-\beta)/2} \Big[1 + \underbrace{\frac{(1-\beta)^2}{24} \ln^2\Big(\frac{F}{K}\Big)}_{\text{分母 2 項}} + \underbrace{\frac{(1-\beta)^4}{1920} \ln^4\Big(\frac{F}{K}\Big)}_{\text{分母 3 項}} + \cdots \Big]} \tag{8.9}$$

$$z := \frac{\nu}{\alpha_0} \big(FK\big)^{(1-\beta)/2} \ln\Big(\frac{F}{K}\Big) \tag{8.10}$$

$$x(z) := \ln\Big(\frac{\sqrt{1 - 2\rho z + z^2} + z - \rho}{1 - \rho} \Big) \tag{8.11}$$

また ATM($F = K$) の場合 次式となる。

$$\sigma_B^{ATM} = \frac{\alpha_0}{F^{(1-\beta)}} \Big\{ 1 + \Big[\frac{(1-\beta)^2}{24} \frac{\alpha_0^2}{F^{(2-2\beta)}} + \frac{\alpha_0 \beta \rho \nu}{4F^{(1-\beta)}} + \frac{2-3\rho^2}{24}\nu^2 \Big] T + \cdots \Big\} \tag{8.12}$$

尚 $F = K$ では $ln(F/K)$=0 であり、式 (8.9) に $ln(F/K) = 0$ を代入すると、ATM の式 (8.12) を得ることが出来る。ただその際、式 (8.10) と (8.11) では $z = 0,\ x(z) = 0$ となるので、式 (8.9) 右辺最初の項は $z/x(z) = 1$ とする点に注意。

図 8.6 では次のパラメータで式 (8.9) を計算させた。

$$\alpha = 0.0306\ , \quad \beta = 0.5\ , \quad \rho = 0.449\ , \quad volvol = 0.728$$

B3 セルに行使価格 0.04 を設定し、B21 セルでブラックのボラティリティ0.69395 を算出。行使価格を変えることで図 8.5 のグラフが出来上がる。

[*7]式 (8.9), (8.10), (8.11), (8.12) は同 [29] の式 (2.17a), (2.17b), (2.17c), (2.18) に対応。

	A	B	C
1	SABRモデルのBlackボラティリティ計算		
2			
3	行使価格:K	4.000%	B列計算式
4	ATM: Fo	0.560%	
5	T	2.000	
6	α	0.031	キャリブレーションで求める
7	β	0.500	固定値
8	ρ	0.449	キャリブレーションで求める
9	volvol	0.728	キャリブレーションで求める
10	1-β	0.500	=1-B7
11	F K	0.000	=B4*B3
12	ln(F/K)	-1.966	=LN(B4/B3)
13	z	-5.72598	= B9/B6 * B11^ (0.5 * B10) * B12
14	x(z)	-2.14778	=LN(((1-2*B8*B13+B13^2)^0.5+B13-B8)/(1-B8))
15	z/x(z)	2.66600	=IF(B3=B4,1,B13/B14)
16	分子 2項	0.00065	=B10^2 /24 * B6^2 / B11^B10
17	分子 3項	0.01023	=B6*B7*B8*B9 / (4*B11^(B10/2))
18	分子 4項	0.03081	=(2-3*B8^2)/24 * B9^2
19	分母 2項	0.12234	=B11^ (0.5*B10)
20	分母 3項	1.04075	= 1+B10^2/24 * B12^2 + B10^4/1920 * B12^ 4
21	Black Vol	0.69395	=B6*(1+(B16+B17+B18)*B5) / (B19*B20) * B15

図 8.6: 式 (8.9) でのブラックボラティリティ計算例

図 8.6 の計算式について、例えば 16 行目以下の“分子 2 項”、“分子 3 項”等は式 (8.9) の“分子 2 項”“分子 3 項”等に対応するので、ここでは説明を省略する。

尚、式 (8.12) の説明で記したように ATM、つまり $F = K$ の場合、$x(z) = 0$ となり、$z/x(z)$ の計算が出来なくなる。その部分を 15 行目の IF 文で $F = K$ の場合、$z/x(z) = 1$ としている。(従って 式 8.12 は未使用)

8.2.2　モデル キャリブレート

図 8.7 は Excel ソルバーによって、キャリブレートを終えた状態のシート例である。前節で使用したパラメータの値

$$\alpha = 0.0306\ , \quad \beta = 0.5\ , \quad \rho = 0.449\ , \quad volvol = 0.728$$

はこのキャリブレートの計算結果である。

キャリブレートの手法は 8.1.3 節 CEV モデルの場合と大きな差は無いので、変更箇所についてのみ説明する。

ベータ β :

- CEV モデルと SABR モデルの違いはボラティリティα が一定かそうでないかの違いだけである点を思い出そう。
- CEV と SABR の共通のパラメータである β は SABR モデルでキャリブレートするのでは無く、対象となっている資産の変動パターンから事前に与えることが一般的となる。

	A	B	C	D	E	F	G	H	I
1	SABRモデル α ρ volvolのキャリブレーション								
2		α＝	0.0306	ρ＝	0.4494		volvol＝	0.7282	
3	行使価格:K	0.0006	0.0031	0.0056	0.0081	0.0106	0.0156	0.0256	
4	ATM: Fo	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056	
5	T	2.000	2.000	2.000	2.000	2.000	2.000	2.000	
6	α	0.031	0.031	0.031	0.031	0.031	0.031	0.031	
7	β(固定)	0.500	0.500	0.500	0.500	0.500	0.500	0.500	
8	ρ	0.449	0.449	0.449	0.449	0.449	0.449	0.449	
9	volvol	0.728	0.728	0.728	0.728	0.728	0.728	0.728	
10	1−β	0.500	0.500	0.500	0.500	0.500	0.500	0.500	
11	F K	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
12	ln(F/K)	2.234	0.591	0.000	−0.369	−0.638	−1.025	−1.520	
13	z	2.27650	0.90870	0.00000	−0.72109	−1.33332	−2.35788	−3.95895	
14	x(z)	1.94760	0.97776	0.00000	−0.60074	−0.95772	−1.37863	−1.81563	
15	z/x(z)	1.16888	0.92938	1.00000	1.20033	1.39218	1.71030	2.18048	
16	分子 2項	0.00532	0.00234	0.00174	0.00145	0.00127	0.00104	0.00081	
17	分子 3項	0.02923	0.01939	0.01672	0.01525	0.01426	0.01294	0.01144	
18	分子 4項	0.03081	0.03081	0.03081	0.03081	0.03081	0.03081	0.03081	
19	分母 2項	0.04281	0.06455	0.07483	0.08207	0.08778	0.09668	0.10942	
20	分母 3項	1.05278	1.00365	1.00000	1.00142	1.00425	1.01097	1.02423	
21	Black Vol	0.89702	0.48497	0.44908	0.48925	0.52792	0.58327	0.64643	
22	market Vol	0.90780	0.46090	0.45300	0.50170	0.53850	0.58420	0.62720	H列計算式
23	差の２乗	0.00012	0.00058	0.00002	0.00015	0.00011	0.00000	0.00037	=(H22-H21)^2
24							平均2乗誤差(MSE)	0.00019	=SUM(B23:H23)/7
25							2乗平均平方根誤差(RMSE)	0.01388	=H24^0.5

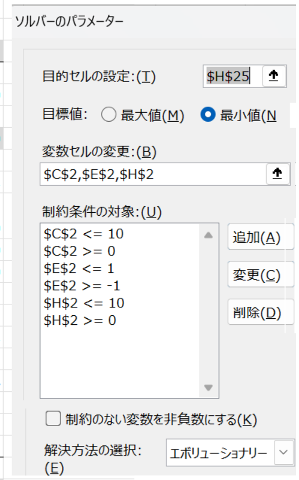


図 8.7: Excel ソルバー エボルーショナリーによるキャリブレート例

- 対象資産がログノーマルなら $\beta = 1$、ノーマルなら $\beta = 0$、中間ならスクェアールートの $\beta = 0.5$ 等。
- 図 8.7 では $\beta = 0.5$ と固定し、キャリブレートの対象から外した。
  - ◇ Brigo and Mercurio [16] では α, ρ, $volvol(\nu)$ は満期・テナー毎にキャリブレートする一方、β は共通としている。
- 8.2.4 節では β の推定方法を紹介する。

残り 3 つのパラメータ α、ρ、$volvol$：

- 残り 3 つのパラメータ α、ρ、$volvol$ に対しては、それぞれ

$$0 < \underbrace{\alpha_0}_{\alpha_t \text{の初期値}}, \quad 0 < \underbrace{volvol}_{\text{正の定数}}, \quad -1 \leq \underbrace{\rho}_{\text{相関係数}} \leq 1$$

  という制約がある。
- 図 8.7 では この制約を “ ソルバーのパラメータ ” の “ 制約条件の対象 ” 欄で設定。[*8]

目的関数： 目的関数は式 (8.5) でよく、計算方法は CEV の図 8.3 G16 セルと同じ。

最適化アルゴリズム：

[*8] α(C2 セル) と $volvol$(H2 セル) は上限 10 (≤ 10) を設定しているが、これはエクセルソルバー機能の制約であり、本来は不要。

- SABR モデルのキャリブレートは 3 つのパラメータを最適化する必要があり、図 8.3 で使用した最適化アルゴリズム“GRG 非線形”では解を見つけ出せなかった。
- **エボリューショナリー**というアルゴリズムを使用すると、図 8.7 の解に到達。
- 一般的には“エボリューショナリー”と言う場合 **Evolutionary Algorithm**(EA) を指し、具体的には Genetic Algorithms、Differential evolution、Particle swarm optimization 等の**メタヒューリスティック手法**での最適化が行われていることを意味する。[*9]

8.2.3 QuantLib, Scipy minimize 関数によるキャリブレート

QuantLib では **sabrVolatility** クラスが SABR モデルのブラックボラティリティを算出する。このクラスの引数は strike、forward、expiryTime と SABR の 4 パラメータで合計 7 つを与える。

sabrVolatility

ql.sabrVolatility(*strike, forward, expiryTime, alpha, beta, nu, rho*)

```
alpha = 1.63
beta = 0.6
nu = 3.3
rho = 0.00002
ql.sabrVolatility(106, 120, 17/365, alpha, beta, nu, rho)
```

QLP ドキュメントに記載された上のコンストラクタの例題 ($strike = 106$, $forward = 120$, $exprityTime = 17/365$ 及び 4 つのパラメータ) をそのまま動かすと 33.28%のブラックボラティリティが算出される (各自確認)。

図 8.7 エクセルで行ったキャリブレートを sabrVolatility クラスと scipy.optimize の **minimize 関数**で行うコードが図 8.8 である。

出力された 4 行目“fun: 0.013880· · · ”と最後の行“[0.03059 0.72825 0.44936])”は図 8.7 の RMSE(H25 セル) 及び α, ρ, $volvol$ と同じであることをまず確認しよう。

[*9] このアルゴリズムに関して、Microsoft 社からの情報を見つけることは出来なかった。メタヒューリスティック手法の入門に関しては Gilli [26] を参照。

```
1  # Sabrカリブレーション
2  from myABBR import * ; from scipy.optimize import minimize
3  # 市場データ
4  STKs = nA([0.06,   0.31,  0.56,   0.81,   1.06,   1.56,   2.56])/100
5  mVol = nA([90.78,  46.09,  45.3,  50.17,  53.85,  58.42,  62.72])/100
6  # パラメータ準備
7  fwdRT,    YR,  beta,  PRMs,                         BNDs                              =¥
8  0.0056,  2.0,  0.5,   [0.1]*3,  [(0.0001,None),(0,None),(-0.9999,0.9999)]
9                            # PRMs=[  alpha,        volvol,       rho ]
10 # 目的関数
11 def calcRMSE(PRMs):
12     vols = [ql.sabrVolatility(ss, fwdRT, YR,
13                           PRMs[0], beta, PRMs[1], PRMs[2]) for ss in STKs]
14     return ((nA(vols) - nA(mVol))**2 ).mean()**.5
15 # カリブレーション
16 RSLT = minimize(calcRMSE, PRMs, method='Powell', bounds=BNDs)
17 print(RSLT,'¥n') ; print('(alpha,volvol,rho):', RSLT.x)
```

```
 message: Optimization terminated successfully.
 success: True
  status: 0
     fun: 0.013880599002214078
       x: [ 3.059e-02  7.282e-01  4.494e-01]
     nit: 5
           (途中省略)
(alpha,volvol,rho): [0.03059 0.72824 0.44936]
```

図 8.8: QuantLib, scipy.optimize でのキャリブレート例

では コードを見て行こう。

2: from scipy.optimize import minimize :

“from モジュール import メソッド”で読み込まれたメソッドはモジュール名を省略できる。(図 7.5 の説明参照)

4, 5: STKs、mVol : 表 8.1 のデータを行使価格用の STKs、ボラティリティ用の mVol に設定。

- **numpy 配列を 100 で割る**場合、配列の各要素が 100 で割られる。(np.array 短縮形 nA は図 2.21 の説明参照)
- 8 行目の `[0.1]*3` のように、Python のリストに対する掛け算とは異なる点に注意。

7, 8: 各種変数の設定 :

PRMs : `[0.1]*3 = [0.1, 0.1, 0.1]` であり、3 つのパラメーター α, $volvol(\nu)$, ρ の初期値を 0.1 に設定。

BNDs : 3 つのパラメーターの上限下限 (bounds) を設定。

- 上限下限の値に記入した数字は含まれる点に注意。
- `None` は上限または下限が無い時の設定。

10〜14: calcRMSE(PRMs)： 最適化の目的関数 RMSE を計算する関数を定義。

- 12 行目の sabrVolatility クラスに 7 行目で初期化した PRMs と他のパラメータを与え、ブラックボラティリティのリスト vols を算出。
- 14 行目で式 (8.5) を計算し RSME を戻す。

16: `minimize(calcRMSE, PRMs, ···)` のコーディングと計算手順：

下図は minimize 関数を説明した Web のスクリーンショットで、必要な引数は `fun`, `x0` の 2 つ。

- `fun` は目的関数, `x0` はカリブレートするパラメータのリスト。
- その他の引数はオプションであり、**`bounds=BNDs`** と **`method='Powell'`** を指定。
- **`method`** によって、最適化の手法を指定する。Scipy では各手法が用意され、その主な一覧は“最適化関数コーディングの注意点”の“4 各種最適化法”(296 ページ)”を参照。

`minimize(fun, x0, args=(), method=None, jac=None, hess=None, hessp=None, bounds=None, constraints=(), tol=None, callback=None, options=None)` [source]

Minimization of scalar function of one or more variables.

Parameters:

fun : *callable*

The objective function to be minimized.

```
fun(x, *args) -> float
```

where `x` is a 1-D array with shape (n,) and `args` is a tuple of the fixed parameters needed to completely specify the function.

x0 : *ndarray, shape (n,)*

Initial guess. Array of real elements of size (n,), where `n` is the number of independent variables.

(この Web アドレスは“最適化関数コーディングの注意点”の“3”を参照)

(この例の具体的な計算手順)

minimize 関数は第 1 引数で指定された calcRMSE 関数に第 2 引数の PRMs を渡し、calcRMSE 関数を実行させる。

- calcRMSE 関数側は minimize 関数から送られてきた PRMs リストを使い、sabrVolatility の引数を構成し、ブラックボラティリティ vols を算出し、minimize 関数に RMSE を戻す。

- RMSE が最小になっていない場合、minimize 関数側は `method='Powell'` に従って、PRMs を修正し、再度 calcRMSE 関数を実行させる。
- RMSE が最小になるまで、上記を繰り返す。

17: RSLT.x： 16 行目の最適化の計算結果は RSLT 変数 (result の略) に設定され、1 つ目の print 文で RSLT をそのまま出力。2 つ目の print 文は `RLST.x` を出力。

- RSLT は辞書型であり、プリント結果の左列 (コロン: の左側) が辞書のキーとなっている。
- ドット演算子を使用し、**辞書. キー**によって、辞書の値へアクセス可能。
- RLST.x はキャリブレートされた 3 パラメータのリストを構成。

(`minimize` 関数の出力キーみにまいずかんすうのしゅつりょくきー) 図 8.8 の RSLT のキーの意味は次の通り。

success: True: 最適化成功。 False: 失敗。

status: 0: 最適化が成功。 1: 解が得られていない可能性が高い。
2: アルゴリズムが収束しなかった。

fun: 目的関数の値。

x: 最適化の解。

nit: 繰り返した回数。

以上が図 8.8 の説明である。次の図 8.9 は 3 行目で RLST.x を sabrVolatility クラスに与え、ボラティリティスマイルを描いている。
若干コードを補足する。

4: nA(cVol)： 短縮形の nA で cVol を numpy 配列としたが、これは cVol の桁数表示をコントロールすることが目的。

8: `label='SABR'`, 12: `ax.legend(title=`⋯`)`：

plot メソッドに label='SABR' を与え、ax.legend と組み合わせることで**凡例** (`legend`) を表示。

- ax.legend() の引数に `title='Powell'` を与えることで、凡例のヘッダー'Powell' を表示。

```
1  # ボラティリティ計算
2  cVol = [ql.sabrVolatility(ss, fwdRT, YR,
3                 RSLT.x[0], beta, RSLT.x[1], RSLT.x[2]) for ss in STKs]
4  print('calc Vols: ', nA(cVol))
5  # SABRグラフ
6  fig, ax = plt.subplots()
7  ax.plot(STKs, mVol, marker='o', label='market')
8  ax.plot(STKs, cVol, marker='^', label='SABR')
9  ax.set_ylabel('Black Vol'); ax.set_xlabel('α,ν,ρ: '+str(RSLT.x))
10 ax.xaxis.set_major_formatter(mtick.PercentFormatter(1,2))
11 ax.yaxis.set_major_formatter(mtick.PercentFormatter(1,0))
12 ax.legend(title='Powell') ; plt.show();
```

```
calc Vols:  [0.89701 0.48497 0.44908 0.48925 0.52792 0.58327 0.64643]
```

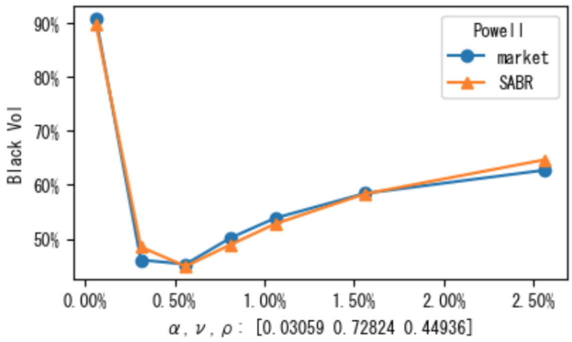


図 8.9: ボラティリティスマイル (市場データと SABR モデル)

10: set_major_formatter(mtick.PercentFormatter(1,2))： x 軸をパーセント表示。(図 1.12 参照)

(最適化関数コーディングの注意点)

以上が SABR モデルを minimize 関数で最適化させる説明となり、非常に長くなったが、最後にコーディングの注意点を記しておく。

(1) sabrVolatility クラスは式 (8.9) を算出するのみのシンプルなクラスで、その他の便利な機能は無い。

- このことは minimize 関数を使う場合にボラティリティを計算する式が在れば、十分なことを意味している。
- 8.3.1 節で説明するノーマルボラティリティの近似式 (8.14) を QuantLib は提供していないが、ノーマルボラティリティのスマイルは式 (8.14) をコーディングすれば済む。

(2) キャリブレートは minimize 関数に計算させる為、エクセル同様 RSME 等の目的関数の用意が必要。

(3) `scipy.optimize` の計算資料 (https://docs.scipy.org/doc/scipy/reference/optimize.html) は良い開示がされている。

- minimize 関数に与える多数のパラメータの役割は Web に情報がある為、生成 AI を利用することを薦める。

(4) minimize 関数が使用できる最適化の計算法 (**最適化法**) は次のように各種提供されている。

(簡便で基本的な 2 つの方法)

`Powell`： 図 8.8 の 8 行目で指定した Powell 法 (Powell's Method)

`Nelder-Mead`： Nelder-Mead 法 (Nelder–Mead method、別名で**ダウンヒルシンプレックス法**、その他の名称も有り)

- Powell 法は Kiusallas[35] を参照。
- Nelder-Mead 法は多くの最適化テキストに解説がある。

(ヤコビアン "勾配ベクトルとも言う" で算出する方法)

`CG`： 共役勾配法 (conjugate gradient method)

`BFGS`： BFGS 法 (Broyden–Fletcher–Goldfarb–Shanno method)

- ヤコビアンの指定が無い場合、scipy の minimize 関数は数値微分を使い、最適化計算を行う。(この場合 Powell、Nelder-Mead 同様のパラメータで済む。図 8.14 参照)

(パラメータの範囲を指定)

`L-BFGS-B`： 図 8.14 の 8 行目で指定した範囲制約付き BFGS 法

`SLSQP`： 等式や不等式の制約も追加可能。(sequential least squares programming)

- CG、BFGS では範囲指定を与えると、Warning される。

(参考文献) 本書では各計算方法の説明は省略する。(興味があれば 各自確認)

- 上記手法の 70%程度をカバーしている入門書としては天谷 [1] や Kiusallas[35] 等がある。
- 天谷 [1] は図が多く、Kiusallas[35] は Python で計算。
- この分野のコードの書き方は生成 AI が便利。

(5) 最適化に関して、その専門家を志さない限り、深入りしないほうが賢明だろう。ただし QuantLib では解の求め方や最適化等で scipy.optimize を多用する為、(最適化の計算方法が判らなくても) 実行可能なコードを学習する価値は大きいと考えよう。

8.2.4 4 つのパラメータと β の推定

この 8.2.4 節では SABR モデルのパラメータ、α, β, ρ, $volvol(\nu)$ の役割を確認後、モデル外で決定される β について考察する。

```
# SABR 4つのパラメータ
plt.rcParams.update({"figure.figsize":[12,8]}) ; fig,ax = plt.subplots(2,2)
# グラフ描写の関数
def drawGraph(xx, yy, ii):
    # ボラ計算
    cVol = [ql.sabrVolatility(ss, fwdRT, YR, *rslt4 ) for ss in STKs]
    uVol = [ql.sabrVolatility(ss, fwdRT, YR, *prmUP ) for ss in STKs]
    dVol = [ql.sabrVolatility(ss, fwdRT, YR, *prmDW ) for ss in STKs]
    # (xx, yy)でグラフの位置  iiはnamePRMリスト中の位置
    ax[xx,yy].plot(STKs, cVol, marker='o', label="base")
    ax[xx,yy].plot(STKs, uVol, marker='^', label="up"  )
    ax[xx,yy].plot(STKs, dVol, marker='.', label="down")
    ax[xx,yy].legend()
    ax[xx,yy].set_title('({}) Base:{:.3f}, Up:{:.3f}, down:{:.3f}'.format(
              namePRM[ii], rslt4[ii], prmUP[ii], prmDW[ii]),fontsize=12)
# シフト幅の設定
rslt4 = np.insert(RSLT.x, 1, beta)        # RSLT.xにbetaを追加
namePRM = ['Alpha','Beta','volvol','Rho']
sftPRM  = [  0.02 , 0.2  ,   0.4   , 0.44]
for ii in range(4):                       # rslt4.copy()は値渡しでの複製
    prmUP  = rslt4.copy() ; prmUP[ii] += sftPRM[ii]
    prmDW  = rslt4.copy() ; prmDW[ii] -= sftPRM[ii]
    xx,yy = divmod(ii, 2) ; drawGraph(xx,yy,ii)
```

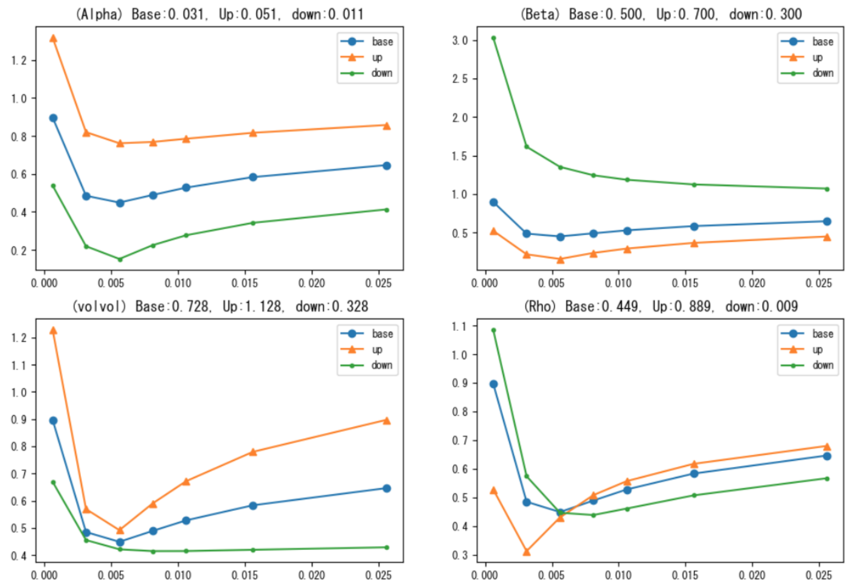


図 8.10: 4 つのパラメータ, 左上:α 右上:β 左下:ρ 右下:$volvol(\nu)$ のシフト例

図 8.10 は各パラメータをシフトさせた場合にスマイルの形状がどのように変わるかを図示した。これらの図から次のことが読み取れるであろう。

(左上) α のシフト： カーブの高さを決定。

(右上) β のシフト： カーブの傾き (スキュー) を決定。

- β が 1 に近いほど、フラットな傾き。0 に近いほど、傾きはきつくなる。
- カーブの高さにも影響。

(左下) $volvol(\nu)$ のシフト： カーブの凸性 (スマイル) を決定。

(右下) ρ のシフト： カーブの傾き (スキュー) を決定。

つまり β は ρ と α の役割と重なっている。そこで 8.2.2 節で説明した通り、一般的に β をモデルの外で決定する。

β の推定法に移る前に図 8.10 のコードに関して補足しておく。

2: `plt.subplots(2,2)`： `plt.rcParams.update("figure.figsize":[12,8])` によって、fig サイズを 12×8 へ拡大し、plt.subplots(2,2) によって、2 行 2 列のグラフを表示させる準備を行う。(rcParams.update は図 1.11 参照)

6〜8: *rslt4, *prmUP, *prmDW： 4 パラメータ (18、21, 22 行で作成) の引数を*印アンパックで準備。

10: ax[xx,yy]、23:`divmod`： 2 行 2 列のグラフを **`ax[行番号, 列番号]`** によって指定。

- 左上が ax[0,0]、右上が ax[0,1]、$\cdots$。
- 23 行目 **`xx,yy = divmod(ii, 2)`** により、ii を 2 で割った商を xx, 余りを yy に設定し、グラフの行番号, 列番号を計算。

17: rslt4=`np.insert(RSLT.x, 1, beta)`： 4 パラメータをまとめて扱う為に numpy insert 関数で RSLT.x[1] の位置に beta を挿入し、rslt4 リストを作成。

20〜23： for ループで１つづつパラメータをシフトさせた変数 prmUP、prmDW を作成し、23 行目の drawGraph でグラフ描写。

(β の推定方法)

β の推定方法を説明した文献は少ないが、ここでは Hansen [30] に記載された方法を紹介しよう。[*10]

まず ATM のボラティリティ (σ_B^{ATM}) は式 (8.12) の第 1 項のみを利用すると、次式のように近似出来る。

$$\begin{aligned} \sigma_B^{ATM} &\simeq \frac{\alpha_0}{F^{(1-\beta)}} \text{ より 両辺に対数をとり、} \\ \ln\bigl(\sigma_B^{ATM}\bigr) &\simeq \ln\alpha_0 + (\beta-1)\times\ln F \end{aligned} \tag{8.13}$$

[*10] 加藤・吉羽 [4] は実証分析から $\beta = 0.6$ を推定。

この式は ATM ブラックボラティリティの対数値がフォワード F の対数値 ($\ln F$) で線形回帰でき、$\ln F$ の係数は $\beta - 1$ であることを表している。この関係を利用し、市場データから β を推定する。[*11]

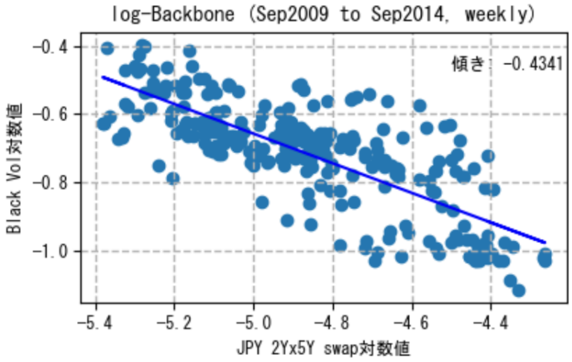


図 8.11: 2 年 ×5 年のボラティリティとスワップレートの線形回帰

図 8.11 では 2 年 ×5 年フォワードスワップレートの対数値 (X 軸) と同スワップションのブラックボラティリティの対数値 (Y 軸) を 2009 年 9 月〜2014 年 9 月までの 5 年間を週ベースでプロットし、式 (8.13) で線形回帰した直線を表示した。右上の“ 傾き:–0.4341 ”が $\beta - 1$ であり、$\beta = 0.5659$ となる。

尚、図 8.11 右側 プロット図のタイトルの **Backbone**(バックボーン) とは Hagan [29]10 ページに記載された用語で“原資産が動いた際の ATM ボラティリティの軌跡 ”を意味する。(この図では対数ベースで描かれている為、“ log-Backbone ” と表示)

8.3 Shifted SABR モデル (マイナス金利対応)

8.2 節では SABR モデルでブラックボラティリティのスマイルを分析した。この節では初めに SABR モデルのノーマルボラティリティ近似式を説明する。ただ現状の QuantLib(version1.32) ではこの計算クラスは無く、別途コーディングする。

次にノーマルボラティリティ近似式をマイナス金利に対応させる為、Shifted SABR モデルを紹介し、マイナスの行使価格を有するノーマルボラティリティのスマイルを算出する。

[*11] CEV モデルの式 (8.4) を使用しても式 (8.13) は成立する。従って CEV モデルのベータ推定も同様の方法で可能。